



Bundle Load

Introduction

There are many different load types in QlikView – each being great for specific purposes. One of these is a bundle load – this is used to include external files, more commonly used for images and sound files. These images/sounds can then be populated within objects in the dashboard running off that data model. This powerful feature can also link to a field in your data model and act dynamically which will be covered in this white paper. Benefits of using a bundle load will be outlined along with the illustration of using relative paths, a form of referencing the data source. The examples used throughout this white paper are constructed using QlikView 12.0 SR5.

Bundle Load Function

Using the 'Bundle' function before a load statement allows you to load in other file types into your data model where the syntax is just like your typical data load – where the first field holds the value of what that file (i.e. an image) will be named as. The second field holds the location of the relevant file. See below an example script where the file location is explicitly stated for loading in country flag images.

```
Flags:
Bundle LOAD * INLINE [
    Country,      Flag
    United States, C:\Documents\QlikSmart\Reference\CountryFlag\United States.png
    United Kingdom, C:\Documents\QlikSmart\Reference\CountryFlag\United Kingdom.png
    Denmark,      C:\Documents\QlikSmart\Reference\CountryFlag\Denmark.png
    Canada,       C:\Documents\QlikSmart\Reference\CountryFlag\Canada.png
];
```

The above script illustrates field one holding the value of the country each flag represents, where the second script holds the exact location of the file itself including the file name and format. Once a reload is performed a table will be held in the data model which includes one field and its values; 'Country'. The name of this table will be of first field, whilst holding a prefix '\$orphan_'.

In the front end, type the below into a text box and change the representation to image.
='qmem://fieldname/fieldvalue'

Field name being the first field in the bundle load, in this case is Country, and an example of field value being Canada – you will find the text box will now illustrate the image.



Bundle Load with Dynamic Functionality

The above functionality is great as it allows you to view an image which can add value to the design of your dashboard. If you loaded in multiple images and wanted to show a specific one based on the values of a field selected, one method would be to use multiple objects placed on top of each other – with the relevant one appearing using the conditional criteria with the layout tab of an object's properties. This can require a lot of resource regarding development time.

Having a dynamic way where the correct image is shown and only one object in the user interface is needed, would be highly beneficial – and there is a way.

In the load script, ensure to name the first field in the bundle load the same name as the field you want it to work dynamically off within your data model. i.e. If you would like the text box to change image based on a selection made in the field: CountryOne, assign that name to the first field within the bundle load. The syntax within text box would need updating too – as shown below.

`=Info(fieldname)` – where the field name here would be 'CountryOne'.

Test this out for yourself. Add a list box for 'CountryOne' and change the value from one to another to see if the results change.

Using Relative Paths

As illustrated in this white paper, explicit paths were used to locate each file within a bundle load. If you were to use this for a production dashboard which may move from a Development server to a UAT server for testing, these locations may need updating (server details in the path). By using relative paths, you are specifying the route of the data source relative to where the dashboard is located – an example shown below.



Figure 1 – Bundle.qvd held in 'Dashboard' and Country flags held in separate png files within 'Countries'

To get from the Bundle.qvw dashboard to the flag png files, you would need to travel back one folder and then into Reference, which holds the Country folder. When using relative paths, this would be denoted as per the below.



..\Reference\Countries\Filename.png – where '..\' refers to going back a folder and Filename are of the image files. If the folder structure is consistent when moving from one server to another, the file locations will not need to be changed within the script – making maintenance less resource heavy.

TIP: Using bundle load allows the more flexibility when previewing images/audio in a dashboard. If an image needed to be replaced and was stored within multiple textboxes, instead of changing the properties of each text box, the image file being replaced would dynamically update all relevant objects – as this file is being read by the dashboard.

When using explicit or relative paths, these can also be specified in a spreadsheet that is loaded into the qvw – shown below.

Country	Flag
United States	..\Reference\CountryFlag\United States.png
United Kingdom	..\Reference\CountryFlag\United Kingdom.png
Denmark	..\Reference\CountryFlag\Denmark.png
Canada	..\Reference\CountryFlag\Canada.png

Figure 2

This allows a business user to maintain the values of the fields and values outside of the script.

NB. Bundle load should be added at the end of your script.

If you have any queries, please contact Ricky Tanna.

